

University of Colorado, Boulder
CU Scholar

Computer Science Undergraduate Contributions

Computer Science

Fall 12-1-2006

Summaries of Wikipedia deletion discussions: a shallow semantic approach

Scott Robert Williams

University of Colorado Boulder

Follow this and additional works at: http://scholar.colorado.edu/csci_ugrad

Recommended Citation

Williams, Scott Robert, "Summaries of Wikipedia deletion discussions: a shallow semantic approach" (2006). *Computer Science Undergraduate Contributions*. Paper 18.

This Thesis is brought to you for free and open access by Computer Science at CU Scholar. It has been accepted for inclusion in Computer Science Undergraduate Contributions by an authorized administrator of CU Scholar. For more information, please contact cuscholaradmin@colorado.edu.

Summaries of Wikipedia Deletion Discussions: A shallow semantic approach

Scott Williams

December, 2006

Abstract

The thesis proposes a method to produce keyword summaries of pro/con debates highlighting the issues the two opposing sides raise in discussing the question.

1 Introduction

1.1 Background

The proliferation of massively distributed editing platforms across the Internet has decreased the barrier to entry preventing average denizens of the Internet from contributing to its content. With the inrush of unskilled and perhaps untrusted users, sites such as Wikipedia (*Wikipedia* n.d.) have struggled to scale their administration practices to cope with the massive number of untrusted edits.

The human scalability issues inherent in the explosion of Wikipedia are many and varied. For example, anybody can create or edit a Wikipedia page. Wikipedia has a volunteer, self-appointed editing community that reviews these edits to make sure nothing inappropriate or contrary to Wikipedia policy has been produced. Users can revert most changes. The exception is that deleting a page cannot be undone. Because the danger to Wikipedia is high if untrusted users were allowed to delete pages on a whim, the power to delete pages is vested in a small class of privileged users called Administrators. As of 2006, there are roughly 2.9 million registered users, of which approximately 1,080 have Administrative privileges.¹

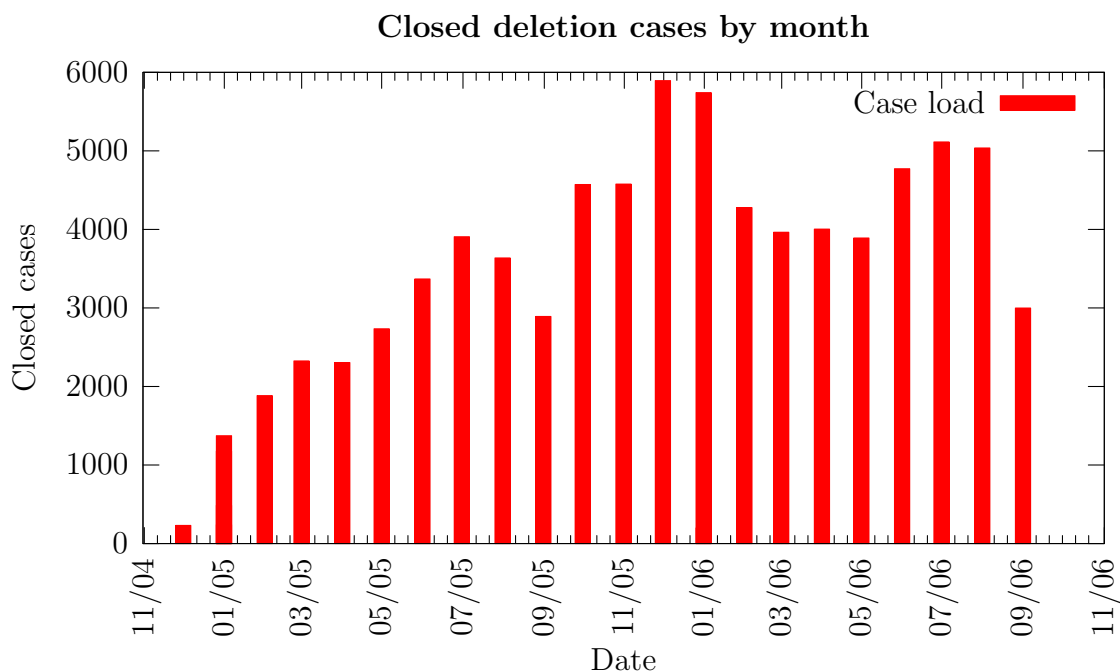
¹<http://en.wikipedia.org/w/index.php?title=Special:Listusers&limit=20&offset=1077&group=sysop>

1.2 The deletion process

When someone discovers an article that should be permanently deleted for any one of a wide and ever-evolving set of criteria, this nominator places a notice on the article to be deleted. Members of the community then discuss whether or not the article should be deleted for a 5–7 day period. A closing administrator reviews the debate to see what the community’s rough consensus is, if it exists. The administrator then reviews previous nominations for deletion that serve as precedent before making the final decision about whether to remove the article.²

Unfortunately for the frustrated and overworked administrator, pages are nominated daily by the tens or hundreds for deletion. Taking into account the accumulated precedent these cases represent is nearly impossible. Over the period from Christmas 2004 to September 19, 2006, the period over which the thesis data ranges, 81,605 nominations were made and voted on.

Figure 1 The volume of closed deletion cases over time shows a growth trend. This growth is likely to continue for the foreseeable future, based on the increasing popularity of Wikipedia.



²http://en.wikipedia.org/wiki/Wikipedia:Articles_for_deletion

1.3 The goal of the project

The purpose of this thesis project is to ease the burden of Wikipedia administrators somewhat. The goal is to summarize deletion discussions to provide at a glance some indication of whether a given deletion discussion would serve as relevant precedent to a case under study. The result of summarization is two sets of keywords—one listing important criteria discussed in votes for deletion, and one listing equivalent criteria discussed in votes against deletion.

2 Previous work

The Message Understanding Conference (Grishman & Sundheim 1996) provides a wealth of work on Information Extraction (Cowie & Lehnert 1996). The problem posed by this thesis is a kind of summary generation, a task well-explored in the MUC literature.

The approach advocated here builds off work in Information Retrieval, specifically the *term frequency—inverse document frequency* keyword selection method explored by Salton (1989). Lin (1998) gives a good background on the information-theoretic considerations behind document similarity metrics.

3 Approach

A typical natural language processing pipeline approach is employed to divide the problem into more tractable pieces. Details are given for how the corpus was collected and cleaned, how the raw corpus was split into cases and votes, and how the votes were categorized according to intent. In further steps, stopword removal (including author signatures—author name, links to the user’s talk page and contributions, and the date of the vote) and stemming increase the quality of the available terms. Finally TF·IDF weights are computed for all the terms occurring in the corpus and keywords for the argument summaries are chosen.

3.1 The corpus

Wget (Niksic 1996) was used to crawl the deletion archives. This produced 640 html files totalling 418 MB, each containing the deletion archives for a single day. These were named with inconsistent English date names according to the title of the source page, reflecting

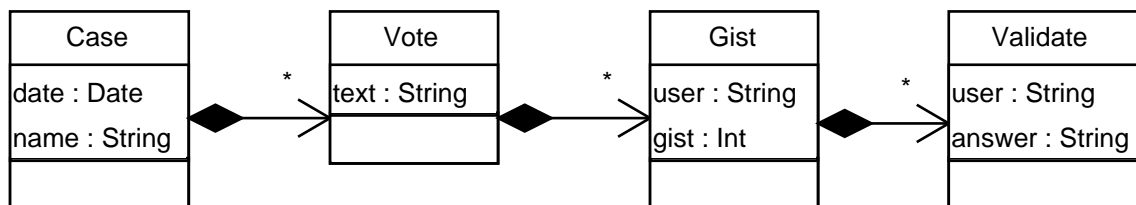
changing archival practices by the Wikipedia community. The files were renamed according to a YYYY-MM-DD scheme to make access easier.

Early practices in the Wikipedia community produced archive pages that were difficult to parse. Archives from 2003 until Christmas 2004 were removed from the corpus to avoid polluting the corpus with badly-extracted cases based on guessing the boundaries between consecutive cases within a given archive file. The effect of this loss is negligible. Archives later than Christmas 2004 are fairly regular, though several of my friends who helped tag the corpus noticed badly-extracted cases—evidence of either tampering with the archives or improperly archived cases.

The next step is to strip out most of the HTML, while at the same time breaking the corpus into cases and votes. Most of the HTML is garbage—the archives are littered with HTML `font` and style abuse that provides no additional semantic information about the text. Some of the HTML, like the `div` tags that indicate the boundaries of a case, are useful. Cases are split into votes by looking for paragraph, list, and definition-making tags. XSLT (*XSL Transformation* (XSLT) n.d.) proved to be the perfect tool for this job.

With the volume of cases so high, they cannot be efficiently dealt with by placing a single case in a single file on the filesystem. I opted to store the cases and vote in an SQLite (Hipp 2002) database—a decision I have come to regret, but it does offer language-neutral access to data, and being able to write queries in SQL made producing charts easy.

Figure 2 The database schema for storing the corpus and tracking annotations. A *gist* represents one user’s decision about which category a vote belongs to. A *validation* represents whether or not one user agrees with the gist chosen by another.



About 1.1 million votes are contained in the database for an average of 13 votes per case. The term *vote* is perhaps misleading, since some of the text excerpts contained in the Vote table are not actual user-submitted votes, but automatically included warnings to refrain from editing the archives, information about the results of the debate, off-topic comments, flames, and incorrectly-parsed excerpts whose meaning cannot be decided.

3.2 Segregating votes

The goal of the project is to produce a keyword summary of *yes delete* votes as well as a separate summary for *no don't delete* votes. The simplest way to achieve this is to segregate the votes into two camps beforehand and summarize them separately, while removing terms that occur with equal frequency in both sets of votes.

Since the deletion discussions are essentially freeform discourse with no enforced requirements about the format of a vote or comment, a reasonable course of action was to sample the database. A crew of about 25 volunteers³ collectively categorized 8,328 votes (0.76% of the corpus). After performing the categorization task, volunteers were asked to rate how well they agreed with categorizations made by others.

Despite the tireless work of my volunteers, too few votes were categorized to use the hand-categorized votes alone in constructing summaries. Because the votes that volunteers categorized were selected randomly, almost zero cases had more than one hand-categorized vote within it. However, the hand-categorized votes provide an evaluation framework for choosing a heuristic to provide better coverage.

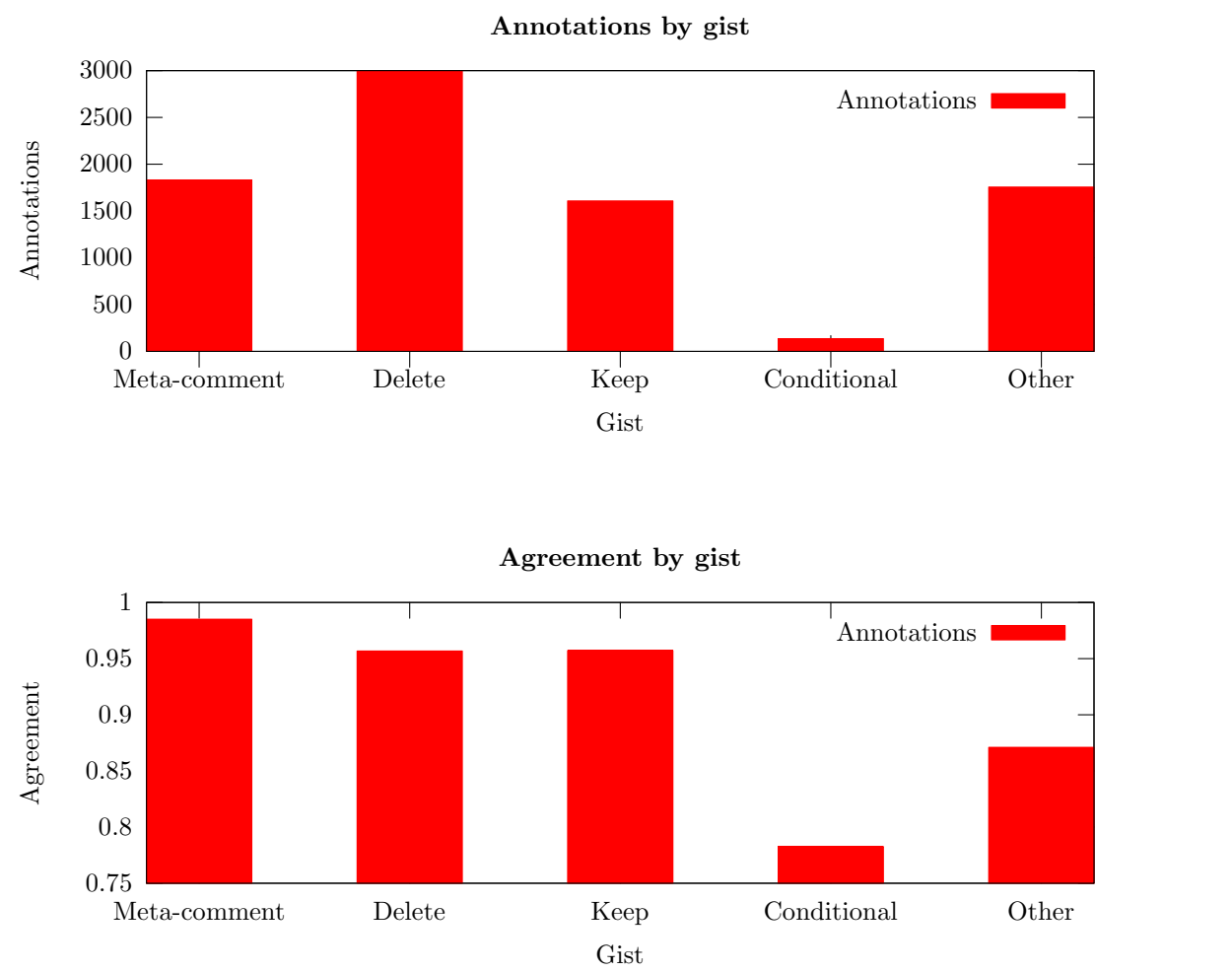
I chose a naïve but effective heuristic using only the beginning of the vote. If the vote began with *delete* then the vote was counted in the *yes delete* category. If the vote began with *keep*, *redirect*, *merge*, *don't delete*, or *speedy keep*, then it was counted in the *no, don't delete* category.

This quick-n-dirty approach increases the number of votes tagged *delete* or *keep* from 4,604 (65.0% *delete*) to 451,814 (65.9% *delete*). This relatively conservative approach to automatic categorization successfully produces 75% of the expected *delete* or *keep* votes for the entire corpus, and the votes categorized in this fashion have a high probability of being correctly categorized—72.6% of hand-categorized *delete* votes begin with the word *delete*, and the heuristic wrongly classifies votes as *delete* only 1.0% of the time.

Likewise, 68% of hand-categorized *keep* votes were automatically identified as *keep* votes using this heuristic, while only 1.4% were automatically identified as *keep* but placed by hand into a different category.

³Thanks to the many volunteers who helped tag the corpus: Austin Bernard, Christine Bennett, Kristin Becker, Randy Williams, Nicci Wolters, Ariel Schutte, Susanne Heckler, Laura Scoggin, Kate Wynant, Kathy Williams, Carrie McDonald, Geoffrey Lehr, Todd Wieck, Lauren Snella, Carrie Geppert, Kim O'Dair, Susan Ganster, Sami Stouffer, Stephanie Wilson, David Buck, Leisa Barrett, Scotty Allen, and Ashly Leder.

Figure 3 Volunteers were instructed to choose the gist of an excerpt of the corpus from this list. The graph shows the resulting distribution of gists. The conditional category seemed to confuse many volunteers.



Meta-comment Each deletion discussion has a few notes that aren’t actually part of the discussion. Typically, there’s warnings against editing the archives and reports of the result of the debate. Anything to the effect of “This discussion is closed, don’t make changes” or “The results of the debate were:” should be put in this category.

“Yes, delete” vote The gist of this kind of text is “remove this article from Wikipedia.”

“No, don’t delete” vote The gist of these comments is “don’t delete the article,” or sometimes “do something

else, like merge, redirect, or clean-up.”

Conditional vote Comments in this class typically take the form “if *condition* then *consequent*, otherwise *alternate*.” Since these votes are predicated on the future, they’re difficult to classify as votes in favor or against deletion. For example “If you can find any references, then keep, otherwise delete.”

Other comment The rest of the comments fall into this category. These comments are neutral or ambiguous with respect to deleting, personal attacks, or not understandable.

Figure 4 A vote that illustrates the difficulty of deciding which words are part of the signature.

Delete. Noetic Null. Eldereft 22:44, 4 August 2005 (UTC)

3.3 Producing potential terms

With the votes split into *keep* and *delete* camps, the next step is to transform votes from English text into bags of terms from which to select keywords.

Due to overzealous HTML pruning during the cleaning phase, not enough context remains in the votes to easily tell where the vote ends and the author’s signature begins. The signatures should be stripped from the text to avoid confusing the term selection algorithm—after all, choosing the name of a debater is probably never good behavior for an algorithm that’s supposed to pick up more objective concerns like *neutrality* or *spam*.

I unfortunately didn’t keep track of where in the original HTML the votes in the corpus came from, so I processed the original HTML a second time with XSLT to produce a list of usernames. However, usernames are unconstrained, so simply pruning the corpus of all the terms occurring in the username list depletes the corpus of usefulness. A better approach is to remove the date and then at most two consecutive usernames—hopefully this catches the talk page link as well as the user’s signature.

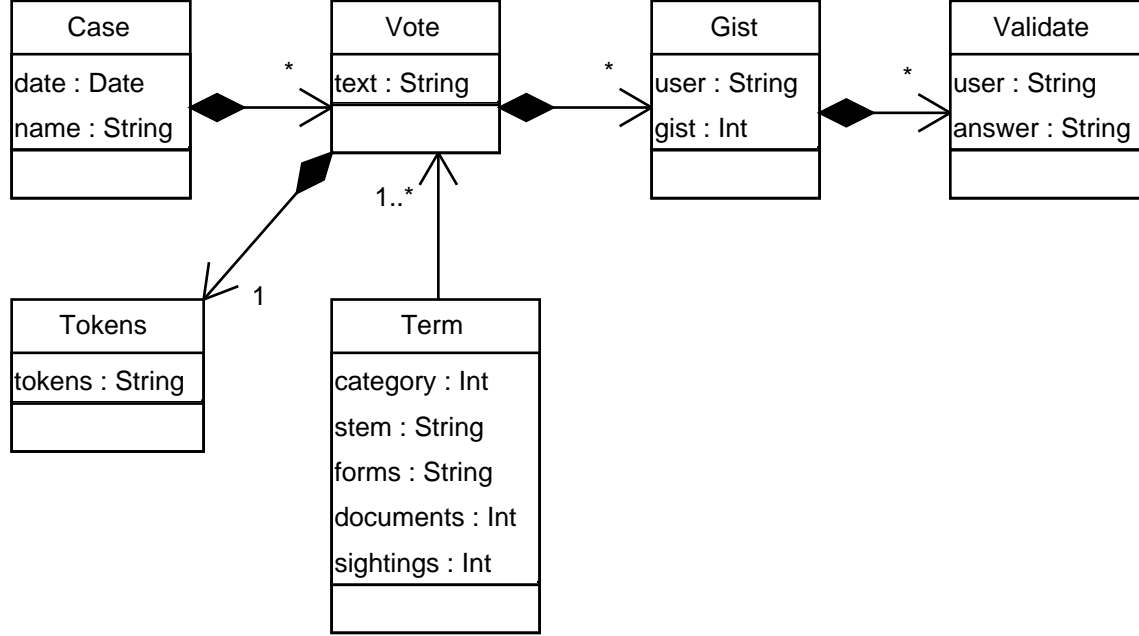
With the signature out of the way, each text fragment is tokenized. Employing a freely-available stopword list from Lextek, superfluous pronouns, auxiliary verbs, and uninteresting adjectives are removed. The tokenized fragment is recorded in the *tokens* table to avoid repeating the work. This step occasionally reduces entire votes to the empty string. About 5.7% of the votes encountered have zero tokens after the stopwords have been removed.

The Wikipedia corpus has a vocabulary of 55–56 thousand words, roughly half of which occur only a single time.

3.4 Background relevance

Once the corpus is tokenized, processing proceeds to the *background relevance* step. Background relevance is computed once for every term in the collection of all *keep* votes, and once for every term in the collection of all *delete* votes. The idea is to weight down words which have a semantic equivalence to *keep* or *delete*—for example, many of the *delete* votes contain the words *speedy* or *strong*, both of which indicate that the author feels very strongly

Figure 5 The database schema after being extended to store tokenized votes and terms. The *Term* table stores background relevance statistics per term per category. If a term occurs in more than one category, the term has more than one entry in the *Term* table.



about deleting the article. These two words are not reasons for deleting an article, they're synonymous with *delete* in this context.

The background relevance w'_i of term i in category c is a weight reflecting the likelihood that term i is an important influence on the outcome of the debate. Suppose the number of votes in category c is V , and v represents the number of votes containing term i at least once. Let n be the number of times term i occurs in all the votes of category c , and N be sum of the terms in all the votes in category c . The terms are stemmed using a Porter stemmer before their scores are computed to increase the conflation of terms with similar meanings.

The traditional TF·IDF weight w_i is computed this way:

$$w_i = \frac{n}{N} \cdot \log \frac{V}{v}$$

The background relevance measure w'_i is computed this way:

$$w'_i = \frac{n}{d}$$

SQLite is missing the log function, so w'_i is a reasonable approximation of the true TF·IDF score.

3.5 Discussion relevance

After the background relevance has been computed for both categories and all their constituent terms, processing proceeds apace to computing discussion relevance. Discussion relevance is the discussion-level cousin of background relevance—a score designed to reflect how important a term is to the given discussion. Discussion relevance is computed separately among categories to determine which terms characterize each category at a local level.

The discussion relevance for a term i in a discussion d ’s category c votes is called w_i'' . Let n_v represent the number of times term i appears in vote v , and let $|v|$ represent the number of terms in vote v .

$$w_i'' = \sum_{v \in c} n_v \cdot w_i' \cdot |v|$$

Multiplying by $|v|$ helps to mitigate the effect of relevance spamming—when one author repeats a term over and over again: “Delete! Stupid, stupid, stupid, stupid!” For example.

3.6 Choosing keywords

Once the discussion relevance has been computed for the terms in the discussion, the top ten terms by w'' -weight are chosen to represent each category. Terms which occur in the top ten of both categories are ignored.

The framework shows promise for delivering these summaries to the community of Wikipedia administrators. Anecdotal reviews of the output indicate it produces summaries of middling quality, but no empirical study of acceptability has been done—the human scaling issues involved in asking volunteers to wade through 300 votes to see how well the summary matches didn’t seem worthwhile to solve at the time.

4 Further work

The output of the algorithm is sensitive to the exact formulation of w'' . It’s not immediately clear whether there is a better formulation which produces higher-quality summaries.

The Wikipedia corpus is incredibly dirty. Unlike the WSJ, misspellings, acronyms, and made up words confound the term selection by artificially increasing the number of unique terms.

Figure 6 Sample output from the program along with the original text. A small selection of 305 keep/delete votes from the original case are presented for comparison.

Keywords for Brian Chase (Wikipedia hoaxer) (35767) on 2005-12-11

Reasons for deleting: notability, edited, vandalized, encyclopedia, hoaxed, person, entire, gone

Reasons for keeping: seigenthaler, notably, redirect, vandals, john, biography, people, chase

- | | |
|--|--|
| <ul style="list-style-type: none"> • Delete. His name can be mentioned in the controversy article. Besides that, he's totally non-notable. Jacoplane 16:32, 11 December 2005 (UTC) • Delete immediately. He is not notable. Just because he hoaxed Wikipedia does not make him notable. If we make an article about him, we have to make an article about every person who gets banned from Wikipedia. Zordrac(talk)Wishy WashyDarwikinian-Eventualist 16:54, 11 December 2005 (UTC) • Delete ridiculous. there are hoaxes every day. what makes this guy more noteworthy than any other wikipedia troll? is anyone going to care about him in a year? i thought not. this is an encyclopedia, not a blog. Derex 19:03, 11 December 2005 (UTC) • Delete Wikipedia is an encyclopedia, and I don't think that pages describing edits should be allowed. Articles get reverted and edited all the time, just because this one got more press there needs to be an entire page about this guy? Sure there can be a snippet in the vandalized article that refutes the false claims, but there does not need to be an entire page that dwells on this edit alone. Leave the guy alone, there is no reason to ruin this person over an edit. Get over it people, you have a encyclopedia that anyone can edit, what did you expect? -Rain 22:34, 11 December 2005 (UTC) • Delete No vandal should gain notability for his actions on wikipedia. - malo(tlk)(cntrbtns) 23:02, 11 December 2005 (UTC) • Delete Avoid self references. Rhollenton 03:22, 12 December 2005 (UTC) • Delete If we were to keep this article, we might as well keep an article on every person mentioned in the news, regardless of how important or unimportant they are. This gentleman, before this incident, was non-notable and will be non-notable after the incident has faded from our memories. There is absolutely no need for an article, no matter how brief it is, to be written about him. • Keep or redirect This entry is essential to preserving the credibility of Wikipedia. This whole situation demonstrates that a self-correcting entity like | <p>Wikipedia can be a reliable source for information and that it does have all the necessary safety measures built in for policing its content.</p> <ul style="list-style-type: none"> • Keep or Merge, but definitely not delete. If anything, the amount of reactions on this VfD should demonstrate some demonstration of notability. The Minister of War(Peace) 20:21, 12 December 2005 (UTC) • Merge or Keep I don't have much to say here, I've only barely ever contributed anything to the Wikipeda. But I think this info should definitely stay in, in some form or another. As others have said, this is an important part of Wikipedia's history. Anyway, I suspect it's important enough that even if you totally delete it, it'll just come back in some form anyway. There's nothing to stop someone from recreating the entry from a locally cached copy of the page, is there? PragmaticallyWyrd 21:07, 15 December 2005 (UTC) • Keep or merge. It happened. It was relatively important. It is our duty to document it.... • Keep. Siegenthaler has said he wanted accountability, and this is evidence that Wikipedia has a certain level of transparency and accountability. Further, this story made national news and will be a notable event in the history of this project. Since this guy got "rewarded" with losing his job and apologizing in person to Siegenthaler innhopes of avoiding litigation, I don't believe this is going to encourage others. Jokestress 16:22, 11 December 2005 (UTC) • Keep because he's notable enough to warrant his own page. The fact is, his identity and name have widely reported. As a result, he's not just another vandal, I'm surprised that some people don't appear to understand this. Whether or not it will encourage other vandals is irrelevant. We should not hide an article just because it may encourage vandalism. The widespread reporting is far more likely to encourage further vandals anyway. Merge with redirect is acceptable but not preferable. He's now notable enough for his own article... Nil Einne 04:36, 13 December 2005 (UTC) • Keep Keep this article as a self-corrective to Wikipedia and warning to future hoaxers. Wikipedia cannot claim to be a reliable source of information unless it opens itself up to self-scrutiny. Kemet 12 Dec 2005 |
|--|--|
-

One way around this problem is to use edit-distance to find stems, which has the added benefit of automatically correcting spelling mistakes (Kantrowitz, Mohit & Mittal 2000).

The shallow semantic approach produces poor keyword summaries for concept-laden *keep* votes. Keep votes in general are 1.5 times longer than *delete* votes, and are additionally more erudite. Single terms fail to capture the essence of the argument. The approach may be saved to some degree by using phrasal salience in addition to single-term TF·IDF to compute background relevance, which would produce multi-word phrases in addition to single terms (Hammouda & Kamel 2002).

Work in indexing by latent semantic analysis (Hofmann 1999) using singular value decomposition of document matrices provides a method for reducing the vocabulary size of the corpus by conflating nearby (in terms of TF·IDF) terms.

An additional step in the pipeline might allow an editor to search for precedent. The user would paste a Wikipedia deletion discussion into the query box. The website would cluster the deletion discussions and retrieve for review the discussions that fall in the query discussion's cluster. This would make it easier to find precedent.

The approach could be generalized to any kind of debate. An interesting case to study would be newsgroups, which present additional challenges of discovering the discourse relationships between posts.

References

- Cowie, J. & Lehnert, W. (1996). Information extraction, *Commun. ACM* **39**(1): 80–91.
- Grishman, R. & Sundheim, B. (1996). Message understanding conference-6: a brief history, *Proceedings of the 16th conference on Computational linguistics*, Association for Computational Linguistics, Morristown, NJ, USA, pp. 466–471.
- Hammouda, K. M. & Kamel, M. S. (2002). Phrase-based document similarity based on an index graph model, *ICDM '02: Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM'02)*, IEEE Computer Society, Washington, DC, USA, p. 203.
- Hipp, D. R. (2002). Sqlite.
URL: *sqlite.org*

- Hofmann, T. (1999). Probabilistic latent semantic analysis, *Proc. of Uncertainty in Artificial Intelligence, UAI'99*, Stockholm.
URL: citeseer.ist.psu.edu/hofmann99probabilistic.html
- Kantrowitz, M., Mohit, B. & Mittal, V. (2000). Stemming and its effects on tfidf ranking (poster session), *SIGIR '00: Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, ACM Press, New York, NY, USA, pp. 357–359.
- Lin, D. (1998). An information-theoretic definition of similarity, *Proc. 15th International Conf. on Machine Learning*, Morgan Kaufmann, San Francisco, CA, pp. 296–304.
URL: citeseer.ist.psu.edu/95071.html
- Niksic, H. (1996). Gnu wget.
- Salton, G. (1989). *Automatic text processing: the transformation, analysis, and retrieval of information by computer*, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA.
- Wikipedia (n.d.). <http://en.wikipedia.org>.
- XSL Transformation (XSLT) (n.d.). W3C Working Draft.
URL: <http://www.w3.org/TR/WD-xslt>